In the Specification

Please amend the following paragraphs of the specification as shown:

[0019] Once the session control 200 allows a session to begin, a user may submit a **Structured Query Language ("SQL")** request, which is routed to the parser 205. As illustrated in Fig. 3, the parser 205 interprets the SQL request (block 300), checks it for proper SQL syntax (block 305), evaluates it semantically (block 310), and consults a data dictionary to ensure that all of the objects specified in the SQL request actually exist and that the user has the authority to perform the request (block 315). Finally, the parser 205 runs an optimizer (block 320), which generates the least expensive plan to perform the request.

[0020] The new set of requirements arising from diverse workloads requires a different mechanism for managing the workload on a system. Specifically, it is desired to dynamically adjust resources in order to achieve a set of per-workload response time goals for complex "multi-class" workloads. In this context, a "workload" is a set of requests, which may include queries or utilities, such as loads, that have some common characteristics, such as application, source of request, type of query, priority, response time goals, etc., and a "multi-class workload" is an environment with more than one workload. Automatically managing and adjusting database management system (DBMS) resources (tasks, queues, **Central Processing Unit ("CPU")**, memory, memory cache, disk, network, etc.) in order to achieve a set of per-workload response time goals for a complex multi-class workload is challenging because of the inter-dependence between workloads that results from their competition for shared resource.

[0027] The system includes a "closed-loop" workload management architecture capable of satisfying a set of workload-specific goals. In other words, the system is an automated goal-oriented workload management system capable of supporting complex workloads and capable of self-adjusting to various types of workloads. The system's operation has four major phases: 1) assigning a set of incoming request characteristics to workload groups, assigning the workload groups to priority classes, and assigning goals (called Service Level Goals or SLGs) to the workload groups; 2) monitoring the execution of the workload groups against their goals; 3) regulating (adjusting and managing) the workload flow and priorities to achieve the SLGs; and 4) correlating the results of the workload and taking action to improve performance. The

performance improvement can be accomplished in several ways: 1) through performance tuning recommendations such as the creation or change in index definitions or other supplements to table data, or to recollect statistics, or other performance tuning actions, 2) through capacity planning recommendations, for example increasing system power, 3) through utilization of results to enable optimizer self-learning, and 4) through recommending adjustments to SLGs of one workload to better complement the SLGs of another workload that it might be impacting. All recommendations can either be enacted automatically, or after "consultation" with the database administrator ("DBA"). The system includes the following components (illustrated in Fig. 4):

1) Administrator (block 405): This component provides a **Graphical User Interface ("GUI")** to define workloads and their SLGs and other workload management requirements. The administrator 405 accesses data in logs 407 associated with the system, including a query log, and receives capacity planning and performance tuning inputs as discussed above. The administrator 405 is a primary interface for the DBA. The administrator also establishes workload rules 409, which are accessed and used by other elements of the system.

2) Monitor (block 410): This component provides a top level dashboard view, and the ability to drill down to various details of workload group performance, such as aggregate execution time, execution time by request, aggregate resource consumption, resource consumption by request, etc. Such data is stored in the query log and other logs 407 available to the monitor. The monitor also includes processes that initiate the performance improvement mechanisms listed above and processes that provide long term trend reporting, which may including providing performance improvement recommendations. Some of the monitor functionality may be performed by the regulator, which is described in the next paragraph.

3) Regulator (block 415): This component dynamically adjusts system settings and/or projects performance issues and either alerts the database administrator (DBA) or user to take action, for example, by communication through the monitor, which is capable of providing alerts, or through the exception log, providing a way for applications and their users to become aware of, and take action on, regulator actions. Alternatively, the

regulator can automatically take action by deferring requests or executing requests with the appropriate priority to yield the best solution given requirements defined by the administrator (block 405).

[0043] In summary the regulator:

a) Regulates (adjusts) system resources against workload expectations (SLGs) and projects when response times will exceed those SLG performance thresholds so that action can be taken to prevent the problem.

b) Uses cost thresholds, which include CPU time, **Input/Output ("IO")** count, disk to CPU ratio (calculated from the previous two items), CPU or IO skew (cost as compared to highest node usage vs. average node usage), spool usage, response time and blocked time, to "adjust" or regulate against response time requirements by workload SLGs. The last two items in the list are impacted by system conditions, while the other items are all query-specific costs. The regulator will use the PSF to handle dynamic adjustments to the allocation of resources to meet SLGs.

c) Defers the query(ies) so as to avoid missing service level goals on a currently executing workload. Optionally, the user is allowed to execute the query(ies) and have all workloads miss SLGs by a proportional percentage based on shortage of resources (i.e., based on administrators input), as discussed above with respect to the two methods for adjusting the allocation of system resources.

[0001] g) Cross-compares workload response time histories (via Query Log) with workload SLGs to determine if query gating through altered **Teradata Dynamic Workload Manager ("TDWM")** ~~TDQM~~ settings presents feasible opportunities for the workload.